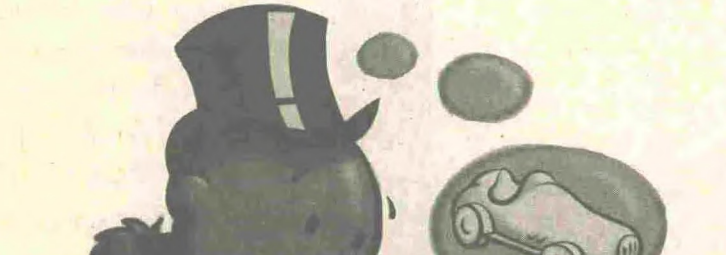
ศรันย์ ไมตรีเวช

VBScript : ใช้อีเวนต์ปรับปรุงการทำงานของฟอร์ม



การป้อนอินพุตอาจทำให้ผู้เข้าเยี่ยมเวบเกิดความรู้สึกราวกับเจ้าของเวบกำลังนั่งอยู่ในคอมพิวเตอร์เพื่อโต้ตอบกับเขาตลอดเวลา

ทางหนึ่งที่จะทำให้เวบเพจของคุณน่าสนใจก็คือ ทำให้ผู้เข้าเยี่ยมเวบเขารู้สึกเหมือนกำลังคุยอยู่กับใครสักคน เปลี่ยนความรู้สึกว่าเขากำลังตั้งหน้าตั้งตาอ่านเอกสารข้อมูลอันแห้งแล้งตามลำพัง ให้กลายเป็นการสนทนาโต้ตอบระหว่างเขากับหน้าเอกสารที่เปี่ยมไปด้วยชีวิตชีวา ถ้าทำได้อย่างนี้ ก็แปลว่าคุณใช้ประโยชน์จากคุณสมบัติความเป็นคอมพิวเตอร์แล้วอย่างเต็มที่

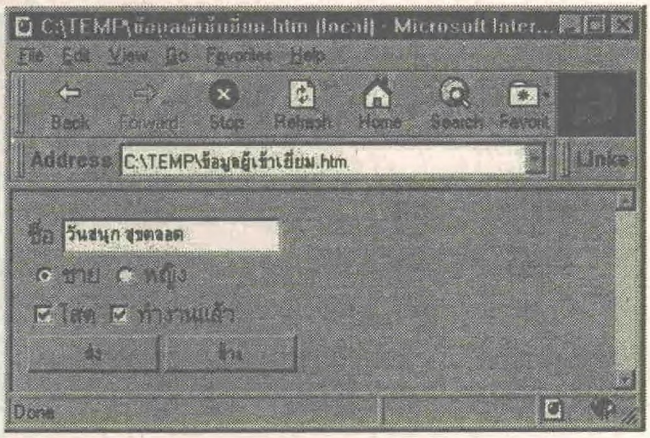
เบื้องต้นของการออกแบบเวบเพจ แน่นอนคุณคิดถึงเนื้อหาข้อมูลว่าเกี่ยวกับอะไร สมควรนำเสนอให้หวือหวาหรือเรียบง่ายสมน้ำสมเนื้อ จากนั้นก็อาจวางแผนว่าจะใช้ลูกเล่นอะไรใน HTML 3.2 เพื่อตีกรอบการนำเสนอให้ดูดีที่สุด ไอ้ที่มากไปก็ทำให้ดูน้อยจะได้ไม่รำคาญตา หรือไอ้ที่น้อยไปก็ทำให้ดูมีน้ำหนักเพิ่มขึ้นสักหน่อย เทคนิคที่ปัจจุบันเจ้าของโฮมเพจใช้กันบ่อยสุดน่าจะได้แก่ Table และรองลงมาคือ Frame เหล่านี้เป็นเรื่องของเอาต์พุตเพื่อเหนี่ยวนำอินพุตบางอย่างจากผู้เข้าเยี่ยม เป็นต้นว่าทำหน้าแสดงผลิตภัณฑ์ให้ดูโดดเด่นน่าจับต้องน่าครอบครองเป็นเจ้าของ เขาจะได้ให้หมายเลขบัตรเครดิตเรามาสั่งสินค้าไปชื่นชม

เมื่อมาถึงขั้นของการทำให้เวบเพจโต้ตอบกับผู้เข้าเยี่ยมได้ แน่นอนต้องนำเรื่อง Form เข้ามาเกี่ยวข้อง จะให้ผู้ใช้ป้อนอินพุต จะให้ผู้ใช้คลิ๊กปุ่มส่งอินพุตมาถึงเรา ล้วนอยู่ในขอบเขตของฟอร์มทั้งสิ้น เมื่อตอนที่แล้วผมกล่าวถึงฟอร์มและอินพุตอย่างง่ายคือ Button หรือที่เราเคยรู้จักในสภาพแวดล้อมวิชวลเบสิกคือ Command Button เพื่อให้คุณคุ้นเคยและรับรู้เกี่ยวกับ VBScript อย่างรวดเร็ว

คราวนี้เรามากล่าวถึงองค์ประกอบในฟอร์มของบราวเซอร์กันทีละตัวครับ ผมจะใช้ชื่อคอนโทรลของวิชวลเบสิกมาเรียกเพื่อให้คุ้นเช่นเดิม

ฟอร์มรับอินพุตจากผู้เข้าเยี่ยมเวบ

กลุ่มคอนโทรลที่อยู่บนฟอร์มนั้นมีจุดประสงค์หลักเพื่อไว้รับอินพุตจากผู้เข้าเยี่ยมเวบ ไม่ว่าจะเป็นอินพุตที่เกิดจากคีย์บอร์ดหรือเม้าส์ คอนโทรลหลักส่วนใหญ่จึงถูกสร้างขึ้นด้วยแท็ก <INPUT> ไปจำแนกชนิดคอนโทรลก็ด้วยแอตทริบิวต์ที่ชื่อ TYPE เท่านั้น ช่วงนี้ผมจะขอกล่าวถึงเหล่าคอนโทรลที่สร้างขึ้นด้วยแท็ก <INPUT> ดังเห็นหน้าตาอินเตอร์เฟสจากรูปที่ 1 เรียงลำดับข้อตามบรรทัดที่เห็นนี้เลย



รูปที่ 1 คอนโทรลรับอินพุตต่าง ๆ ที่เกิดขึ้นได้บนฟอร์ม

1. Text Box ใช้คีย์บอร์ดป้อนข้อความ ซึ่งอาจมีลักษณะเป็นสารธรรมดา ตัวเลข หรือพาสเวิร์ด ความยาวของข้อความอาจมีเพียงประโยคสั้นหนึ่งประโยค หรืออย่างมากผูกกันหลายประโยคเป็นหนึ่งย่อหน้า เพราะลักษณะการแสดงช่องรับบรรทัดเดียวบอกผู้เข้าเยี่ยมอยู่ในตัวว่า "ไม่ต้องยาวนัก"

เราสร้าง Text Box ด้วยการกำหนดชนิดให้เป็น TEXT เช่นในที่นี้คือ

```
<INPUT TYPE=TEXT NAME="VisitorName">
```

และเพื่อทำให้คำว่า "ชื่อ" ปรากฏกำกับข้างหน้าช่องรับ ก็เพียงเขียนลงไปข้างหน้าแท็ก <INPUT> เท่านั้น เพราะบราวเซอร์จะแสดงอักษรตามที่เราเขียนขึ้นมาลอย ๆ อยู่แล้ว

เมื่อผู้เข้าเยี่ยมเวบกรอกข้อมูลในช่องนี้ เช่น เขาชื่อ "วันสนุก สุขตลอด" แล้วทำการส่งฟอร์มมายังอีเมล์แอดเดรสของเรา เราจะเห็นข้อมูลที่ได้มาเป็น VisitorName=วันสนุก สุขตลอด จะเห็นว่า VisitorName นั้นเป็นค่าของแอตทริบิวต์ NAME ซึ่งเรากำหนดให้เป็นอะไรก็ได้ แม้แต่ภาษาไทย ขอให้ได้เอาต์พุตที่กำกับเนื้อข้อมูลแล้วอ่านรู้เรื่องเป็นพอ

2. Option Button ใช้เม้าส์คลิ๊กเพื่อกำหนดเลือกอะไรขึ้นมาหนึ่งเดียวจากกลุ่ม เช่นในที่นี้บอกเพศว่าผู้เข้าเยี่ยมเขาเป็นชายหรือหญิง เมื่อเกิดขึ้นบนฟอร์มเดียวกัน หมายความว่าทำงานร่วมกัน

เราสร้าง Option Button ด้วยการกำหนดชนิดให้เป็น RADIO เช่นในที่นี้คือ

```
<INPUT TYPE=RADIO NAME="Gender" VALUE="ชาย">ชาย
<INPUT TYPE=RADIO NAME="Gender" VALUE="หญิง">หญิง
```

จะสังเกตว่าค่าของแอตทริบิวต์ NAME นั้นเหมือนกัน ที่นี้ผมตั้งไว้เป็น Gender เพื่อให้ได้เอาต์พุตออกมาเพียงค่าเดียวเป็น M หรือ F เช่นเมื่อนายวันสนุกคลิ๊กเลือก "ชาย" ก็หมายความว่าเมื่อเขาส่งฟอร์มมาถึงเรา เราจะเห็นเขียนว่า Gender=ชาย

อย่าสับสนว่าทำไมผมเขียนชายและหญิงซ้ำซ้อนกันนะครับ "ชาย" กับ "หญิง" ข้างนอกแท็ก <INPUT> นี้เอาไว้กำกับ Option Button ที่ปรากฏในบราวเซอร์เพื่อให้ผู้เข้าเยี่ยมเขาหมายรู้เฉย ๆ ส่วน "ชาย" กับ "หญิง" ที่เป็นค่าของแอตทริบิวต์ VALUE นั้น เอาไว้บ่งให้เราทราบว่าผู้เข้าเยี่ยมเขาคลิ๊กเลือกปุ่ม Option ไหน

3. Check Box ใช้เม้าส์คลิ๊กเพื่อกำหนดเลือกหนึ่งหรือมากกว่าหนึ่ง จากกลุ่มรายการที่อาจสัมพันธ์หรือไม่สัมพันธ์กันก็ได้ เช่นในที่นี้เราให้ผู้เข้าเยี่ยมระบุสถานภาพสมรสของตนเอง และบอกด้วยว่าเรียนจบหรือยัง ซึ่งไม่เกี่ยวกัน เขาอาจไม่เลือกอะไรเลย อาจเลือกตัวใดตัวหนึ่งหรืออาจเลือกทั้งสองดังที่เห็น

เราสร้าง Check Box ด้วยการกำหนดชนิดให้เป็น CHECKBOX เช่นในที่นี้คือ

```
<INPUT TYPE=CHECKBOX NAME="Status" VALUE="โสด">โสด
<INPUT TYPE=CHECKBOX NAME="Occupation" VALUE="ทำงานแล้ว">ทำงานแล้ว
```

ถ้าผู้เข้าเยี่ยมเวบคลิ๊กเลือก Check Box ตัวใด ค่าของ NAME และ VALUE จะถูกส่งมาถึงเรา ถ้าไม่เลือกก็ว่างไป เช่นตัวอย่างนี้นายวันสนุกคลิ๊กเลือกทั้งสองกล่อง ดังนั้นเราจะเห็นอีเมล์มาถึงเราว่า

```
Status=โสด
Occupation=ทำงานแล้ว
```

4. Submit Button ใช้เม้าส์คลิ๊กเพื่อส่งอินพุตทั้งหมดที่ปรากฏในฟอร์มขณะนั้นไปที่ใดที่หนึ่งตามที่ระบุไว้ในแท็ก <FORM> เราสร้างขึ้นด้วยการกำหนดชนิดให้เป็น SUBMIT เช่นในที่นี้คือ

```
<INPUT TYPE=SUBMIT VALUE=" ส่ง ">
```

หากไม่ใส่แอตทริบิวต์ VALUE ไว้ในแท็กก็ไม่เป็นไร แต่จะเห็นปุ่มปรากฏคำว่า SUBMIT แทน คุณไม่จำเป็นต้องเขียนแอตทริบิวต์ NAME ไว้ เพราะปุ่มนี้เอาไว้ส่งข้อมูลทั้งฟอร์มอยู่แล้ว จึงมีได้เพียงหนึ่งเดียว หากเขียน NAME ก็จะเห็นข้อมูลเพิ่มขึ้นหนึ่งบรรทัดเปลืองเนื้อที่เปล่า

5. Reset Button ใช้เม้าส์คลิ๊กเพื่อล้างอินพุตทั้งหมดออกจากฟอร์ม เพื่อพร้อมรับใหม่จากผู้เข้าเยี่ยม เราสร้างขึ้นด้วยการกำหนดชนิดให้เป็น RESET เช่นในที่นี้คือ

```
<INPUT TYPE=RESET VALUE=" ล้าง ">
```

อย่างที่กล่าวไว้ในตอนที่แล้วว่าคอนโทรลรับอินพุตจะเกิดขึ้นลอย ๆ ในบราวเซอร์ไม่ได้ ต้องมีฟอร์มขึ้นมารองรับ ซึ่งเราก็สามารถสร้างฟอร์มขึ้นด้วยแท็ก <FORM> นั่นเอง ในเวบเพจหนึ่ง ๆ อาจมีได้หลายฟอร์ม ปุ่ม Submit เป็นของฟอร์มไหนก็ส่งฟอร์มนั้นไปยังจุดหมายปลายทางที่กำหนดไว้

การกำหนดจุดหมายปลายทางของอินพุตที่อยู่ในฟอร์มทั้งหมดนั้น จะทำกันด้วยการระบุ URL ซึ่งโดยหลักการแล้วก็หมายความว่าเราอาจรับอินพุตผ่านโปรโตคอลเช่น mailto: หรือ http: ซึ่งถ้าเป็น Netscape แล้วเราจะส่งฟอร์มได้ด้วยโปรโตคอลทั้งสองชนิด แต่น่าเสียดาย Internet Explorer ส่งฟอร์มได้ด้วยโปรโตคอล http: เท่านั้น

การส่งฟอร์มผ่านโปรโตคอล http: หมายความว่าเราให้บราวเซอร์มันเข้าถึงไฟล์ที่สามารถรับข้อมูลในฟอร์มไปประมวลผลได้ ซึ่งในปัจจุบันก็หมายถึงการใช้สคริปต์ CGI นั่นเอง สำหรับสคริปต์ CGI นี้เป็นคนละเรื่องกับสคริปต์ HTML, JavaScript และ VBScript มันเป็นด่านหน้ารับอินพุตส่งต่อไปให้เซอร์ฟเวอร์ประมวลผลโต้ตอบอีกต่อ คุณต้องเรียนรู้ที่จะเขียนมันขึ้นมาด้วยภาษาเช่น Pearl ซึ่งนิยมใช้สร้างสคริปต์ CGI กันมาก

เพื่อทดลองจริงกันเดี๋ยวนี้ ไม่ต้องรอเรียนภาษา CGI ให้เป็นเสียก่อน ทางลัดที่สุดก็คือใช้สคริปต์ซึ่งคนอื่นเขียนไว้อยู่แล้ว อย่างเช่นถ้าคุณมีโฮมเพจฟรีที่จีโอซิตีส์ (http://www.geocities.com/) ก็จะมีสิทธิ์ใช้สคริปต์ซึ่งเขาสร้างไว้แล้วสำเร็จรูปทันที ชื่อ mail.pl (ถ้าเห็น .pl แสดงว่าเขียนด้วยภาษา Pearl) ซึ่งอยู่ในไดเรคทอรี /cgi-bin/homestead/

วิธีชี้ให้บราวเซอร์เข้าถึงสคริปต์ CGI ผ่าน URL นั้น เราใช้แอตทริบิวต์ของแท็ก <FORM> 2 ตัว ได้แก่

1. ACTION ใช้กำหนด URL ที่จะเข้าถึงสคริปต์ CGI โดยตรง เช่นถ้าหากต้องการทดลองโดยส่งอินพุตในฟอร์มมาถึงผม ก็เขียนว่า ACTION="http://www.geocities.com/cgi-bin/homestead/mail.pl?dangtrin"

2. METHOD สำหรับแอตทริบิวต์นี้ จะมีค่าได้ 2 อย่าง คือ POST และ GET โดยทั่วไปเราใช้ POST ได้ตลอด ซึ่งหมายถึงวิธีการส่งอินพุตไปเป็นต่างหากจาก URL ส่วน GET จะแปะอินพุตเข้าไปในส่วนท้ายของ URL ซึ่งอาจเกิดข้อผิดพลาดเพราะความยาวของ URL มีขีดจำกัดอยู่

ในตัวอย่างตอนนี้เป็นเรื่องของการส่งอินพุตไปถึงอีเมล์แอดเดรส ฉะนั้นเลือก METHOD เป็น POST และ ACTION เป็น URL ชี้ไปหาสคริปต์ที่จีโอซิตีส์ เขียนได้ว่า

```
<FORM METHOD=POST ACTION="http://
www.geocities.com/cgi-bin/homestead/
mail.pl?dangtrin">
```

คุณอาจเปลี่ยน ?dangtrin ในตอนท้ายเป็น ? แล้วตามด้วยชื่อทะเบียนของใครหรือของคุณเองในจีโอซิตีส์ก็ได้ (หากยังไม่ขอโฮมเพจฟรีที่นั่นก็แนะนำให้ขอเสีย จะได้มีโฮมเพจส่วนตัวบนเซอร์ฟเวอร์เร็วและถาวรไว้ศึกษาทดลองอะไรต่ออะไรได้ โดยเฉพาะมีสิ่งอำนวยความสะดวกสำเร็จรูปหลาย ๆ อย่างเช่นสคริปต์ CGI นี้ มิฉะนั้นการส่งฟอร์มปกติจะเห็นอินพุตเรียงติดกันเป็นพืด)

เมื่อเขียนโค้ดเพื่อสร้างเป็นไฟล์ HTML ทั้งหมดใน Notepad จะเห็นดังรูปที่ 2 สังเกตด้วยว่าผมใช้แท็ก
 ซึ่งหมายถึง Line Break เพื่อทำให้ออบเจ็กต์อยู่ต่างบรรทัดกัน

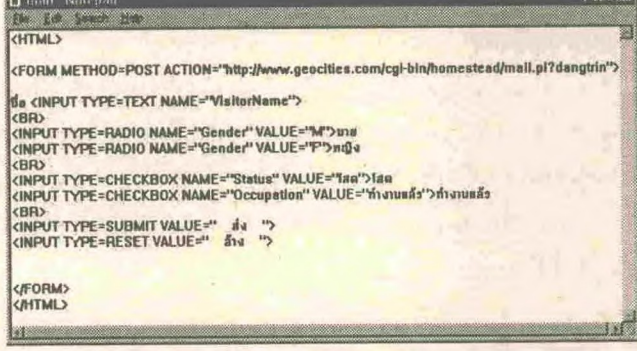
แค่นี้พอแล้วแน่หรือ

ถ้ามองดูทั้งหมดนี้แล้วก็เหมือนทุกอย่างสมบูรณ์แบบ เราพร้อมจะรับอินพุตจากผู้เข้าเยี่ยมกันแล้ว ซึ่งหากเขากรอกข้อมูลครบก่อนส่ง สิ่งที่เราซึ่งเป็นเจ้าของเวบจะได้รับก็คืออินพุตข้อมูลส่วนตัวจากผู้เข้าเยี่ยมดังประสงค์

แต่ถ้าเขาไม่กรอกข้อความอะไรเลย ไม่ให้อินพุตอะไรบนฟอร์มเลยล่ะ? นั่นแหละครับปัญหา เพราะเราคงไม่อยากเสียเวลาเปิดอีเมล์ที่ว่างเปล่า นำไปวิเคราะห์วิจัยหาประโยชน์จากสถิติไม่ได้

ด้วยกลไกของ HTML เราไม่อาจดักหรือทักท้วงการกระทำชนิดนี้จากผู้ใช้ขี้เล่น ชอบคลิ๊กปุ่มสนุก ๆ อย่างมากที่สุดอาจป้อนค่าดีฟอลต์ให้ไว้เป็น <INPUT TYPE=TEXT NAME="VisitorName" VALUE="Anonymous"> ซึ่งจะเห็นใน Text Box ปรากฏคำว่า Anonymous แต่แรก และหากไม่แก้อะไรตอนคลิ๊กส่ง เราก็จะได้ VisitorName=Anonymous มา เป็นอันรู้กันว่าผู้เข้าเยี่ยมเขาไม่เต็มใจเปิดเผยตัว และในเมื่อไม่อยากเผยตัวก็ไม่น่าคลิ๊กปุ่มส่งให้เป็นภาระแก่ระบบสื่อสารทำไม ด้วย VBScript เราสามารถเขียนโค้ดขึ้นมาสกัดกั้นการส่งฟอร์มของผู้เข้าเยี่ยมประเภทนี้ได้ครับ

คราวที่แล้วผมแนะนำให้คุณรู้จักกับอีเวนต์ onClick ซึ่งผูกติดอยู่กับปุ่ม Button คราวนี้จะให้เห็นการประยุกต์ใช้กับการตรวจสอบความ



รูปที่ 2 โค้ด HTML ซึ่งเขียนเสร็จสมบูรณ์พร้อมถูกบราวซ์

สมบูรณ์ในการส่งฟอร์มบ้าง บอกไว้ก่อนว่าการตรวจสอบอินพุตเป็นไปได้หลายทาง แต่ทางเลือกที่ผมยกตัวอย่างนี้ตั้งใจให้คุณเห็นการประยุกต์ใช้ออบเจ็กต์ในบราวเซอร์เป็นหลัก

คอนเส็ปต์ในการเขียนสคริปต์ขึ้นมาใช้ในครั้งนี้ก็คือ แทนที่จะใช้ปุ่ม Submit ซึ่งมีหน้าที่หลักเป็นการส่งฟอร์มแน่ ๆ เราเปลี่ยนเป็นใช้ปุ่ม Button ธรรมดา ซึ่งอาจทำหน้าที่อะไรก็ได้ เช่นคราวที่แล้วเมื่อคลิ๊กมัน ผมก็ให้มีแสดงเมสเสจบ็อกซ์ "สวัสดีจ้า!" แสดงแล้วก็แล้วกัน ไม่มีการส่งหรือรับข้อมูลต่อเนื่องไปไหนอีก

ดังนั้นอันดับแรกที่เราจะเปลี่ยนก็คือชนิดของอินพุตรูปปุ่ม จาก TYPE=SUBMIT เป็น TYPE=BUTTON แบบนี้

```
<INPUT TYPE=BUTTON VALUE="ส่ง">
```

เมื่อผู้เข้าเยี่ยมเขาคลิ๊กปุ่มชนิด Button ดังกล่าวแล้ว เราต้องให้มีการดำเนินการบางอย่างผ่านอีเวนต์ Click ของปุ่ม Button นั่นเอง ถึงตรงนี้พยายามคิดย้อนไปหาแบบแผนการเขียนโค้ดของวิชวลเบสิกดั้งเดิมนะครับ เราทราบกันดีว่าเมื่อพูดถึง "อีเวนต์" โดยทั่วไปก็คือพูดถึงโพรซีเดอร์ที่มาตอบรับการกระทำอย่างใดอย่างหนึ่งเม้าส์หรือคีย์บอร์ดของผู้ใช้นั่นเอง โค้ดที่อยู่ในโพรซีเดอร์ตอบรับเหตุการณ์หนึ่ง ๆ จะต้องอ้างถึงชื่อออบเจ็กต์ และพรอพเพอร์ตี้ส์หรือเมธอดของออบเจ็กต์เป็นหลัก เพื่อให้เกิดอะไรบางอย่างขึ้นโต้ตอบกับผู้ใช้

ยกตัวอย่างเช่น เมื่อคลิ๊กปุ่ม Ok เราอาจตรวจดูว่าใน TextBox มีข้อมูลอะไรหรือไม่ ถ้าไม่มีก็บอกให้ผู้ใช้ทำให้มีเสีย แต่ถ้ามีตามเกณฑ์ก็เอาลงดาต้าเบสเลย ลองดูเป็นเค้าว่าเราเคยเขียนโค้ดในวิชวลเบสิกธรรมดาอย่างไร

```
Sub cmdOk_Click ()
    If Text1.Text = "" then
        MsgBox "กรุณาป้อนข้อมูลก่อน"
    Else
        Data1.Recordset.Fileds("fstEntry").Value = Text1
    End If
End Sub
```

การเขียนโค้ดด้วย VBScript ก็ใช้หลักอ้างชื่อออบเจ็กต์และพรอพเพอร์ตี้ส์ของออบเจ็กต์เช่นกัน ในที่นี้ออบเจ็กต์ที่เกี่ยวข้องและเราต้องอ้างชื่อมีอยู่แค่ 3 ตัวเท่านั้น คือ

1. ชื่อฟอร์ม เราจำเป็นต้องใช้ชื่อฟอร์มเป็นหลักอ้างอิงออบเจ็กต์ที่อยู่ในมัน ต่างจากวิชวลเบสิกเดิมที่เราอาจเขียนเพียง Text1.Text = "" ได้เลย เพราะรู้ว่า Text1 มันอยู่บนฟอร์มปัจจุบันอยู่แล้ว ในตัวอย่างนี้ผมจึงต้องตั้งชื่อฟอร์ม จะตั้งให้เป็นอะไรก็ได้ เช่น Information ก็เขียนลงที่แท็ก <FORM> ว่า

```
<FORM NAME="Information" METHOD="..."
ACTION="...">
```

นอกจากความสำคัญในการใช้ชื่อฟอร์มเป็นหลักอ้างอิงออบเจ็กต์อินพุตอื่นที่อาศัยอยู่ภายในตัวมัน เรายังจะใช้เมธอด Submit ของออบเจ็กต์ฟอร์มนี้แหละเป็นตัวส่งฟอร์ม ฉะนั้นชื่อฟอร์มจึงเป็นสิ่งแรกที่ขาดไม่ได้เลย

2. ชื่อ TextBox ข้อมูลใน TextBox นี่แหละที่เราอยากตรวจดูเสียก่อนว่าผู้เข้าเยี่ยมเขากรอกหรือเปล่า และอย่างที่ทราบว่าเราไม่มีชื่อเช่น Text1 โดยดีฟอลต์เหมือนอย่างในวิชวลเบสิกธรรมดา จึงต้องตั้งชื่อให้มันเสีย อันนี้ของเดิมเราก็ตั้งไว้แล้วคือ VisitorName

```
<INPUT TYPE=TEXT NAME="VisitorName">
```

3. ชื่อปุ่ม Button หากปราศจากชื่อของปุ่ม Button เราก็สร้างอีเวนต์ Click ของปุ่มไม่ได้ ลองนึกถึงวิชวลเบสิกเก่า จะเห็นว่าทุกอีเวนต์ที่ผูกติดกับออบเจ็กต์จะต้องขึ้นต้นด้วยชื่อออบเจ็กต์เสมอ

```
<INPUT TYPE=BUTTON VALUE="ส่ง" NAME="CheckInput">
```

เมื่อได้ชื่อเสียงเรียงนามของออบเจ็กต์ให้อ้างอิงแล้ว คราวนี้ก็มาถึงการสร้างโพรซีเดอร์เพื่อรองรับอีเวนต์ Click ของปุ่ม Button เราสามารถเขียนโค้ดเพื่อตอบรับอีเวนต์ได้ 3 แบบใน VBScript

1. เขียนแบบ Event-Procedure ซึ่งก็เหมือนกับวิชวลเบสิกดั้งเดิม ดังนี้

```
<INPUT TYPE=BUTTON VALUE="ส่ง" NAME="CheckInput">
<SCRIPT LANGUAGE="VBScript">
    Sub CheckInput_onClick
      If Information.VisitorName.Value="" Then
            MsgBox "กรุณาป้อนชื่อก่อนส่ง"
       Else
            Information.Submit
       End If
    End Sub
</SCRIPT>
```

คุณจะเห็นว่าผมดึงชื่อของปุ่ม Button ซึ่งกำหนดไว้เป็น CheckInput มาตั้งชื่ออีเวนต์รับเหตุการณ์ Click (ใน VBScript ต้องเขียนว่า onClick) ฉะนั้นเมื่อมีการคลิ๊กปุ่ม CheckInput ก็จะเกิดการเรียกอีเวนต์ CheckInput_onClick ทันที

เพื่อเช็คดูว่า TextBox ชื่อ VisitorName มันว่างเปล่าหรือไม่ เราเขียนโค้ดบรรทัดแรกของอีเวนต์เป็น If Information.VisitorName.Value="" Then ตามหลักการเข้าถึงออบเจ็กต์ของออบเจ็กต์โมเดลคืออ้างถึงของใหญ่ก่อนแล้วตามด้วยของเล็ก สำหรับ Value นั้นเป็นพรอพเพอร์ตี้ส์ของ VisitorName เทียบเท่ากับพรอพเพอร์ตี้ส์ Text ของ TextBox ในวิชวลเบสิกเดิม

หากตรงตามเงื่อนไข คือ VBScript พบว่าผู้เข้าเยี่ยมยังไม่ได้ใส่ข้อความลงไปใน TextBox ก็จะทำให้เกิดเมสเสจบ็อกซ์เตือนด้วยโค้ด MsgBox "กรุณาป้อนชื่อก่อนส่ง" ซึ่งอันนี้เราเคยใช้มาแล้วในตอนก่อน

ส่วนถ้าหากไม่ตรงตามเงื่อนไข กล่าวคือผู้เข้าเยี่ยมเขาป้อนอินพุตเป็นอันดี เราก็ดำเนินการส่งฟอร์ม โดยใช้ Submit ซึ่งเป็นเมธอดของฟอร์ม อย่างที่เห็นเขียนคือ Information.Submit นั่นเอง

2. เขียนแบบ Inline Event Handler การเขียนแบบนี้เราไม่สร้างโพรซีเดอร์รับอีเวนต์ไว้ต่างหากเหมือนอย่างแบบแรก แต่แทรกตัวรับอีเวนต์ (Event Handler) ไว้ในแท็กเลยทีเดียว เสมือนว่ามันเป็นแอตทริบิวต์หนึ่งของแท็ก

คุณจะเห็นว่าเมื่อเราต้องมีโค้ดไว้รองรับอีเวนต์ยาว ๆ การเขียนแบบนี้คงไม่เหมาะแน่ การเขียนแบบแทรกอีเวนต์ไว้ในแท็กเหมาะกับการสั่งให้ VBScript ปฏิบัติงานสั้น ๆ เช่นสร้างเมสเสจบ็อกซ์ "สวัสดี" เดี่ยว ๆ เขียนได้ดังนี้

```
<INPUT TYPE=BUTTON onClick='MsgBox "สวัสดี" '
LANGUAGE="VBScript">
```

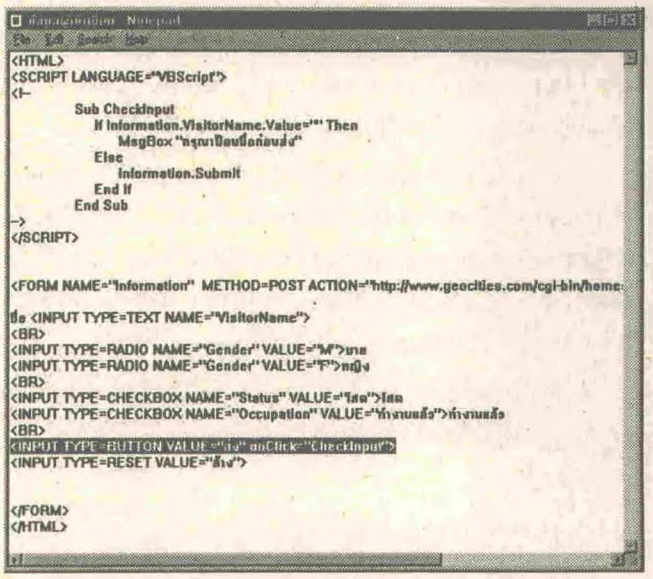
ขอให้สังเกตการใช้เครื่องหมาย ' ไว้หุ้มโค้ดในอีเวนต์ด้วยนะครับ คุณคงเห็นความแตกต่างอย่างชัดเจนกับแบบแรก นั่นคือเราไม่ต้องตั้งชื่อโพรซีเดอร์ เราระบุชื่ออีเวนต์ที่ต้องการลงในแท็ก <INPUT> โดยตรงทีเดียว และใช้เครื่องหมาย = ในการสั่งเสียให้เกิดการปฏิบัติงานอย่างใดอย่างหนึ่งขึ้นตามต้องการ

และที่ผมต้องแทรก LANGUAGE="VBScript" รวมไว้ในแท็กด้วยก็เพราะว่าผมจะไม่ใช้แท็ก <SCRIPT> เลย คุณจะเห็นว่า HTML ไม่มีฟังก์ชัน MsgBox หากเราไม่บอกให้บราวเซอร์รู้ว่านี่เป็นภาษา VBScript บราวเซอร์ก็จะฟ้องว่าเกิดความผิดพลาดหรือไม่ทำงานตามที่เราต้องการ

ความจริง Inline Event Handler เป็นที่นิยมมาก ในตัวอย่างนี้เราอาจเขียนให้ onClick มันไปเรียกโพรซีเดอร์ซึ่งเขียนดักไว้แล้วล่วงหน้าเพื่อหลีกเลี่ยงความรุงรังของแท็ก <INPUT> ก็ได้ ดังที่ผมแสดงตัวอย่างในรูปที่ 3 คือเรียกโพรซีเดอร์ CheckInput ด้วย onClick

3. เขียนแบบ For/Event อันนี้จะกลับเป็นตรงข้ามกับ Inline Event Handler กล่าวคือเราจะนำอีเวนต์และคำสั่งทั้งหมดไปไว้ในแท็ก <SCRIPT> เลยทีเดียว เช่นในตัวอย่างนี้ ถ้าผมตั้งชื่อปุ่มเป็น CheckInput จะเขียนได้ว่า

```
<SCRIPT LANGUAGE="VBScript" FOR="CheckInput" EVENT="onClick">
<!-
        If Information.VisitorName.Value="" Then
            MsgBox "กรุณาป้อนชื่อก่อนส่ง"
        Else
            Information.Submit
        End If
->
</SCRIPT>
```



```
<HTML>
<SCRIPT LANGUAGE="VBScript">
<!-
        Sub CheckInput
          If Information.VisitorName.Value="" Then
            MsgBox "กรุณาป้อนชื่อก่อนส่ง"
          Else
            Information.Submit
          End If
        End Sub
->
</SCRIPT>

<FORM NAME="Information" METHOD=POST ACTION="http://www.geocities.com/cgi-bin/home-

ชื่อ <INPUT TYPE=TEXT NAME="VisitorName">
<BR>
<INPUT TYPE=RADIO NAME="Gender" VALUE="M">ชาย
<INPUT TYPE=RADIO NAME="Gender" VALUE="F">หญิง
<BR>
<INPUT TYPE=CHECKBOX NAME="Status" VALUE="โสด">โสด
<INPUT TYPE=CHECKBOX NAME="Occupation" VALUE="ทำงานแล้ว">ทำงานแล้ว
<BR>
<INPUT TYPE=BUTTON VALUE="ส่ง" onClick="CheckInput">
<INPUT TYPE=RESET VALUE="ล้าง">

</FORM>
</HTML>
```

รูปที่ 3 การใช้ Inline Event Handler เพื่อเรียกโพรซีเดอร์อีกต่อ

ทิ้งท้าย

สรุปแล้วตอนนี้เราศึกษาเกี่ยวกับการใช้อีเวนต์มาช่วยทำให้ฟอร์มซึ่งเป็น HTML ปกติให้สมบูรณ์ขึ้น คุณจะเห็นว่าหากเรามีแต่ความรู้ HTML อย่างเดียว ก็จะสร้างเวบเพจโต้ตอบกับผู้เข้าเยี่ยมได้ไม่สมบูรณ์

อย่าลืมเข้าไปเยี่ยมเวบไซท์ VBScript ของไมโครซอฟท์ที่ http://www.microsoft.com/vbscript นะครับ ขุมข้อมูลและเอกสารอ้างอิงที่จำเป็นอยู่ที่นั่นเยอะมาก พบกันฉบับหน้าครับ สวัสดี